

This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

1. (Original) A system comprising:
a plurality of servers organized into one or more failover groups and over which data is partitioned, each server usually processing client requests for data of a respective type and processing the client requests for data other than the respective type for other of the plurality of servers within a same failover group when the other of the plurality of servers within the same failover group are offline; and,
a master server managing notifications from one or more clients and from the plurality of servers as to whether servers are offline, the master server verifying whether a server is offline when so notified, and where the server has been verified as offline, so notifying the plurality of servers other than the server that has been verified as offline.
2. (Original) The system of claim 1, further comprising a database storing data responsive to client requests of any respective type and which has been partitioned over the plurality of servers, each server caching the data stored in the database responsive to client requests of the respective type.
3. (Original) The system of claim 2, wherein each server further temporarily caches the data stored in the database responsive to client requests other than the respective type when the other of the plurality of servers within the same failover group are offline.
4. (Original) The system of claim 1, wherein the one or more failover groups consists of one failover group, such that the plurality of servers are within the one failover group.
5. (Original) The system of claim 1, further comprising one or more clients sending requests to the plurality of servers.
6. (Currently Amended) A system comprising:
a plurality of servers organized into one or more failover groups, each server usually

processing client requests of a respective type and processing the client requests other than the respective type for other of the plurality of servers within a same failover group when the other of the plurality of servers within the same failover group are offline; and,

a database storing data responsive to client requests of any respective type and which is partitioned for caching over the plurality of servers, each server caching ~~eaching~~ the data stored in the database responsive to client requests of the respective type, each server also temporarily caching the data stored in the database responsive to client requests other than the respective type when the other of the plurality of servers within the same failover group are offline.

7. (Original) The system of claim 6, further comprising a master server managing notifications from one or more clients and from the plurality of servers as to whether servers are offline, the master server verifying whether a server is offline when so notified, and where the server has been verified as offline, so notifying the plurality of servers other than the server that has been verified as offline.

8. (Original) The system of claim 6, wherein the one or more failover groups consists of one failover group, such that the plurality of servers are within the one failover group.

9. (Original) The system of claim 6, further comprising one or more clients sending requests to the plurality of servers.

10. (Currently Amended) A computer-readable medium having instructions stored thereon for execution by a processor to perform a method comprising:

determining whether a data server is in a failover mode;
in response to determining that the data server is not in the failover mode, sending a request to the data server;
determining whether sending the request was successful;
in response to determining that sending the request was unsuccessful,
entering the failover mode for the data server;
notifying a master server that sending the request to ~~[[the]]~~ one of ~~[[the]]~~ a plurality of

data servers, was unsuccessful;

determining a failover server; and,

sending the request to the failover server, capable of processing requests for partitioned data of a respective type and partitioned data other than its respective type.

11. (Original) The medium of claim 10, the method initially comprising determining the data server as one of a plurality of data servers to which to send the request.

12. (Original) The medium of claim 10, the method initially comprising in response to determining that sending the request was unsuccessful, repeating sending the request to the data server for a predetermined number of times, and entering the failover mode for the data server if sending the request for the predetermined number of times was still unsuccessful.

13. (Original) The medium of claim 10, the method further comprising in response to determining that the data server is in the failover mode, determining whether the data server has been in the failover mode for longer than a predetermined length of time; and,
in response to determining that the data server has not been in the failover mode for longer than the predetermined length of time, sending the request to the failover server.

14. (Original) The medium of claim 13, the method further comprising in response to determining that the data server has been in the failover mode for longer than the predetermined length of time, sending the request to the one of the plurality of data servers;
determining whether sending the request was successful;
in response to determining that sending the request was unsuccessful, sending the request to the failover server;
in response to determining that sending the request was successful, exiting the failover mode for the data server; and,
notifying the master server that sending the request to the data server was successful.

15. (Original) A method for performance by a server, comprising:
receiving a request from a client;
determining whether the request is of a type usually processed by the server;
in response to determining that the request is of the type usually processed by the server,
processing the request;
in response to determining that the request is not of the type usually processed by the
server, determining whether a second server that usually processes the type of the request is
indicated as offline;
in response to determining that the second server that usually processes the type of the
request is indicated as offline, processing the request;
in response to determining that the second server that usually processes the type of the
request is not indicated as offline, sending the request to the second server;
in response to determining that sending the request was unsuccessful,
processing the request; and,
notifying a master server that the second server is offline.

16. (Original) The method of claim 15, further comprising receiving indication from
a master server that the second server is online.

17. (Original) The method of claim 15, further comprising receiving indication from
a master server that the second server is offline.

18. (Original) A computer-readable medium having instructions stored thereon for
performing the method of claim 15.

19. (Currently Amended) A machine-readable medium having instructions stored
thereon for execution by a processor of a master server to perform a method comprising:
receiving a notification that a server may be offline;
contacting the server;
determining whether contacting the server was successful;

in response to determining that contacting the server was unsuccessful, marking the server as offline; and,

notifying a plurality of servers, capable of processing requests for partitioned data of a respective type and partitioned data other than its respective type, other than the server marked as offline that the server is offline.

20. (Original) The medium of claim 19, the method further comprising periodically checking the server that has been marked as offline to determine whether the server is back online.

21. (Original) The medium of claim 20, wherein periodically checking the server that has been marked as offline comprising:

contacting the server;

determining whether contacting the server was successful;

in response to determining that contacting the server was successful, marking the server as online; and,

notifying the plurality of servers other than the server marked as online that the server is online.